Towards a new generation of Irrigation Decision Support Systems–Irrigation Informatics?

Car, N.J. [1,2,3], Christen, E.W.[2,3,4], Hornbuckle, J.W. [2,3] and Moore, G.[1]

[1] Department of Civil and Environmental Engineering, The University of Melbourne, Victoria, Australia
[2] Griffith Research Laboratories, CSIRO Land & Water, Griffith, NSW, Australia
[3] CRC for Irrigation Futures
[4] Adjunct Assoc. Prof. University of New England, Armidale, NSW, Australia
Email: nicholas.car@csiro.au

Keywords: *Decision support systems, Web 2.0, irrigation, informatics, irrigation scheduling, Web services, XML, AJAX, Sensor Web Enablement*

EXTENDED ABSTRACT

This paper proposes a new generation of Decision Support Systems (DSS) that leverages Web Services and Web 2.0 technologies to allow for new possibilities in the areas of irrigation decision support.

A new classification system for existing DSS, based on their 'network paradigm' is presented with current systems being placed in categories 1 to 5. Category 1 DSS are those with no networking abilities and are typically represented by desktop applications. Category 2 (C2) are those with direct network links to local equipment, such as sensors in a paddock. Category 3 (C3) DSS use local area networks to access data from such sources as databases and networked sensors. C4 DSS use large, proprietary and purpose-built, networks, such as SCADA networks, to collect data as well as using resources available to C3 DSS. C5, use the internet to access multiple instances of the resources available to C4 DSS. We present some examples of DSS in each of these categories for illustration.

A further category, 6, is proposed here that uses new internet software technology to extend DSS functionality into uncharted waters. Technologies such as extensible Mark-up Language (XML) and Web Services are proposed to allow DSS to provide different types of support to users at many different levels, to allow for the addition of User Defined Data Sets (UDDS) and to utilise the power of machine-to-machine communications over the internet.

We suggest how potential DSS, using some of the technologies mentioned here, may help counter the poor uptake of DSS in Australian agriculture by addressing one of its supposed root causes: that of the lack of user customisation. We propose that in addition to this, a category 6 DSS may be used in a way that no DSS has currently been used and that is in irrigation benchmarking. Further to this: we suggest how a C6 DSS used for irrigation support may present usage metrics for use by 3rd parties, such as water supply companies.

We then propose back-end architecture for a C6 DSS that utilises technologies such as XML-based Web Services, live, online databases and data fusion to bring together and interpret data from distributed providers. We relate how flexible back-end architecture may allow DSS to provide very customizable decision support and how sophisticated networking may be used to generate benchmarking data.

Next we look at how new approaches to interface design using recent 'Web 2.0' technologies, such as AJAX, provide the tools needed by developers to create DSS front ends that can effectively use the DSS back-ends discussed above.

INTRODUCTION

The concept of computerised DSS has been around since the late 1970's and early attempts to classify them focussed on their architecture (Sprague 1980). Five architectural components have typically been used for classification and they are 1. the database management system, 2. the model base management system, 3. the knowledge engine, 4 the user interface and 5. the user (Gachet and Haettenschwiler 2003). Classification in this way gives no clue as to what a particular DSS does. Recently classification has been undertaken using the DSS user's 'assistance method' (Power 2007) and categories in this classification are

1. Model driven, eg. use mathematical, economic or other models
2. Communication driven, eg. Microsoft's Net Meeting designed for group decision support
3. Data-driven, eg. singular or multiple sources to a user to assist them in decision making
4. Document driven i.e. storage and processing technologies for document retrieval and analysis
5. Knowledge driven DSS. – uses expert-derived rule base or expert system reasoning, and presents 'knowledge' to users, rather than just data

DSS in agriculture typically present as model, data and knowledge-driven DSS.

A new classification method is presented here: classification by networking paradigm. It is more related to both classifications by 'assistance method' and by architecture but groups DSS by the types of information that they are able to provide to the DSS user.

DSS related to natural resource management in agriculture have been used to choose crop types, set planting dates and determine fertiliser application rates. For irrigation DSS, some provide 'tactical' support that typically helps the user decide when and with how much water to irrigate – an example for sugarcane is WaterSense (Inman-Bamber, Webb et al. 2006) – while others provide 'strategic' decision support that helps users to plan irrigation systems and make long-term irrigation decisions. An example such, used for assisting in on-farm water storage design is DAMEA$Y (Lisson et al., 2003). For 'tactical' irrigation DSS, support is often derived from meteorological data, meteorological predictions and crop modelling. The software systems and concepts proposed here relate only to 'tactical' irrigation DSS.

Despite many DSS having been created for use specifically in Australian agriculture, there has been very poor uptake (Hayman 2004). Users for any type of agricultural DSS rarely number more than a few hundred (Inman-Bamber and Attard 2005) and our research indicates that many do not run for more than a few 'test' seasons. Reasons for this are thought to relate to the idea that although DSS provide good scientific advice, the are not able, or perceived to be able, to provide 'real world' support (Hayman 2004). Such 'real world' support would be support that is based on all of the factors that decision makers use, not just scientific biophysical factors. These factors may be economic or social, or of some other nature but are certainly not viewed as trivial by the agriculturalist. Our own research in the irrigation scheduling DSS area suggests that factors such as the price of electricity to run irrigation pumps outweighs at least some of the benefits of 'scientifically optimum' irrigation scheduling which would see pumps run at peak power price times.

In recent years, since approximately 2001, a new paradigm in internet technologies, termed 'Web 2.0' (Forrest 2006) and exemplified by websites and other internet applications that allow users to perform tasks that involve extensive interaction with remote computing resources but with limited technical knowledge required, has emerged. It is providing unparalleled access for non-technical people to server resources, databases and other networked information providers.

Such access could allow irrigators to choose their own data sources for decision support, rather than relying on a predetermined set and, if the total set of data sources available for them to choose from included both biophysical and non-biophysical data sets, then they may achieve better 'real world' support.

1. DSS CLASSIFICATION

1.1. The Network Paradigm

Classification by networking paradigm focuses on the abilities of a DSS to present data and information from different resources to a user and also how that data is presented. The word 'networking' is used in both technical and conceptual senses. Classification in this way places existing DSS in the following categories:

1. **None** – standalone desktop application.
2. **Single Link** – desktop applications that collect data from a single machine, logger, or sensor.
3. **LAN** - desktop/intranet-based DSS with information only from local area network (LAN) resources such as other computers,

local network databases and local network sensors.
4. **Enterprise Network** – DSS using large, proprietary and purpose-built, networks, such as (Supervisory Control and Data Acquisition) SCADA networks, to collect data as well as the resources available to C3 DSS.
5. **Internet** – use the internet to access multiple, possibly remote, instances of the resources available to C4 DSS and purpose-built data sets presenting information over the internet.

Figures 1 and 2 show schemas of DSS categories 1 to 5.

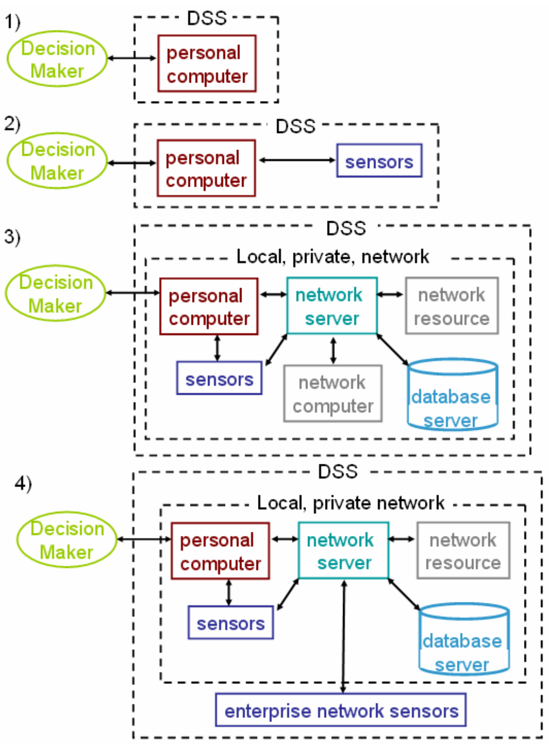

Figure 1. Schema of Category 1- 4 DSS

An example of C1 DSS is the comprehensive crop management tool APSIM 5.3 by APSRU (http://www.apsim.info/apsim/) which models many different factors affecting crop performance using data entered into and stored on the DSS user's personal computer. irriMAX™ (http://www.sentek.com.au/products/irrimax.asp?lang=en) by Sentek is an example of a C2 DSS that presents soil moisture information from field sensors to a PC user. An example of a C3 DSS is 'Probe for Windows' by Research Services New England, which presents soil moisture data from multiple, different probes connected to via a network on a desktop PC. C4 DSS would likely be seen on large corporate farms and urban irrigations system. An example is 'ET Drive' by the South Australian-based company Micromet which uses local evapotranspiration (ET) values, connected via a radio link, in conjunction with database information to control urban irrigation systems. An example of a C5 DSS is WaterSense, a sugarcane irrigation scheduling tool developed by the CSIRO. (Inman-Bamber, Webb et al. 2006). This DSS uses the internet to present its user interface to irrigators as well as using the internet to download remotely calculated ET data.

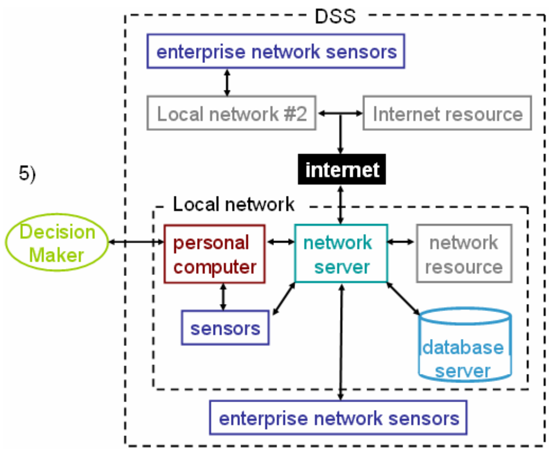

Figure 2. Schema of Category 5 DSS

The authors propose a further category of DSS – category 6 *Interoperably connected* – to be used to describe future DSS that use Web 2.0 and Semantic Web technologies. Figure 3 shows the schema of a category 6 DSS.

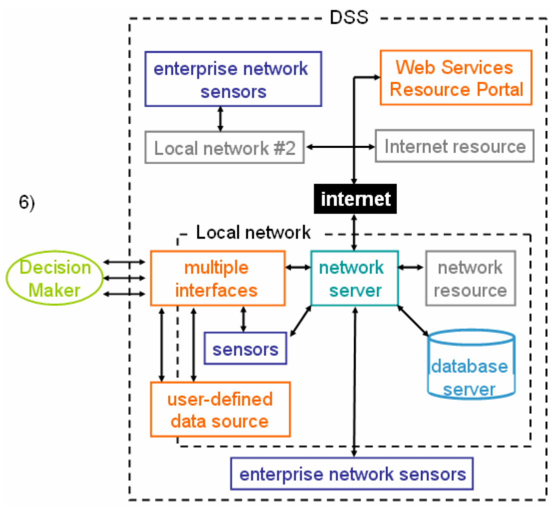

Figure 3. Schema of a Category 6 DSS

The three features that distinguish a C6 DSS from a C5 DSS are new DSS features, **multiple interfaces**, **user-defined data sources** (UDDS) and **Web Services resources**. These three features are made possible by new technologies in internet and software engineering and result in new forms of networking, thus deserving a new category in the 'network paradigm' classification.

1.2. Category 6 Features

Multiple interfaces

Traditionally DSS front-ends have been restricted to desktop programs and internet browsers viewed on PCs. There is at least one example of a mobile computer-based, model-driven, irrigation DSS (Hornbuckle, Christen et al. 2005) that is a C1 DSS on a mobile computer but there are no examples of C2 – 5 DSS that use interfaces other than PCs and none whatsoever that use multiple physical interfaces.

Having a DSS with a primary PC based interface and secondary mobile interfaces will allow for greater flexibility in the DSS's use. In this way any irrigator may view comprehensive advice on a PC in the morning, before going into the paddocks, and then view a simpler version of the same or updated advice later in the field on a mobile phone or SmartPhone at the time when they are most likely to act upon it by doing something like turning on irrigation pumps.

In addition to past DSS being restricted to using a single physical interface, most have also been restricted to delivering support in one mode to one type of user. For example WaterSense only allows users to receive scheduling advice based on a pre defined set of input parameters. It does not allow users with varying degrees of scientific and IT understanding to receive decision support in other ways. It also requires that training of potential users be put in place before use. Potentially a DSS could present very simple decision support information to an untrained user and then allow them to 'opt in' for further modes of more complex decision support.

User-defined Data Sets

A UDDS is envisaged to be a data set, such as a local or remote database, or data source, such as a local soil moisture probe or local automatic weather station, that is added to the pool of data sources used by the DSS by the unassisted DSS user. The key is that the UDDS would be added *after design time*, so the user would be able to add datasets and data sources that the user values that were not specifically catered for by the DSS designers. This is contrasted to the current situation where, even if it were possible for a user to add a new data set to their instance of a DSS after design time, and mostly this is not possible, significant 3rd party involvement (perhaps expensive software consulting) would be needed. Allowing UDDS could have the effect of allowing users to customise the support they receive and thus make it more applicable to their particular situation

Web Services Resources

Capitalised 'Web Services' refer to a software system designed to facilitate machine-to-machine interactions over the internet. One may see Web Services in operation where one has a desktop 'widget' program that communicates with Amazon.com and each day updates a display of the last 10 books published on a particular topic. Web Services are realised as programmatic functions, also known as methods, hosted on a remote server machine that can be accessed by a client application. This is similar to a web server's web pages being accessed by a client using an internet browser but with far more possibilities for the types of data delivered and presentation. The aspects of Web Services technology that set them apart form other internet technologies are those relating to service description and discoverability which allow Web Services to be used by a client machine without the need for human involvement during setup or operation.

The concept of Web Services resources is similar to that of UDDS and would only differ in location of the data provider (remote-via-internet for Web Services resources and local for a UDDS) and that Web Service-enabled data sets could be relevant and available to many DSS users.

2. NEW DSS BACK-END SYSTEMS

2.1. Architecture for C6 DSS

For the proposed C6 DSS, back-end architecture needs to be able to provide:

a. The ability to create different subsets of the total decision support information set. A particular subset would be chosen based on the application used to view decision support.
b. The ability to cope with new, unforseen, data sets and sources relevant to decisions by processes also unforseen by DSS designers.
c. The ability to 'mix and match', at least to some extent, the data sources selected by the users to be used in generating decision support.

With the advent of widespread Internet Protocol (IP) interoperable communications, such as 3rd generation (3G) cellular networks, it is now easier to present DSS information on mobile devices than before, albeit with additional carrier's fees. Additionally other internet-to-wireless services, such as the cellular phone Short Messaging Service (SMS) gateways and the Multimedia Messaging Service (MMS) that use SMS with internet web pages, offer the possibility for DSS to use mobile non web-based data delivery channels. 3G networks and Web Service protocols also allow 'widget' programs on many platforms to present data gathered via the internet in many different ways including cellular phone Java applications.

Great advances in software programming frameworks, such as Microsoft's .NET and the open source Java platform Eclipse, allow back-end code used for PC-based programs to easily be ported to mobile devices such as SmartPhones. The authors are currently testing .NET-based mobile font-ends to DSS. Some of their work can be viewed at http://irrigateway.net/dev/mobile.

There are significant technical obstacles that need to be overcome in order to create a DSS that exhibits characteristics b. and c. The main obstacle is that a DSS needs to understand the format of the data presented to it which, in the case of a range of potential data sources or sets is a very large task. A second obstacle is how a DSS may then use such data, once it has understood it. Two methods to address the first obstacle are now proposed.

The first involves writing software 'drivers' for each of the possible resources that a user may potentially add to their DSS. There is precedence for this, for example Research Services New England's 'Probe for Windows' implements drivers for many (they claim practically all) of the soil moisture probes available in Australia. Since there is a limited range of potential biophysical data sources available to DSS users, this may be possible for them. It may also be possible to write drivers for most of the remote biophysical data sets available for DSS use in Australia, for example ET readings from Queensland's Dept. of Natural Resources and Water's remote interpolated ET service known as SILO project. This method is not comprehensive, will always lag behind new data set establishment and requires much effort on behalf of the DSS designers. The authors believe it to be of limited use.

A second method that could be used to overcome the technical obstacle of DSSes understanding data formats would be the widespread standardisation of data source service description and data interchange formats. If data sources and their data sets are available and presented in a standardised way, their addition to a DSS's reasoning engine could be much simplified.

The Open Geospatial Consortium (OGC) is working on project called Sensor Web Enablement (SWE) which is a group of standards and protocols "specifying interoperability interfaces and metadata encodings that enable real time integration of heterogeneous sensor webs into the information infrastructure."(Open Geospatial Consortium 2007). An implementation of a C6 DSS could allow a user to connect any standards-compliant data source so that broad use of standards as SWE would offer much data source and data set choice available to the DSS user.

One of the potential aspects of SWE is that it allows data sources to be 'discoverable' thus allowing machines, such as DSS, to find and use them without human involvement. An example situation using a version of SWE could be seen if a DSS user wished to run a decision support advice generation scenario using up-to-date evapotranspiration (ET) information (used for irrigation scheduling from SILO and compare the results to the same scenario using up-to-date ET data from a local, web-enabled, Automatic Weather Station (AWS). They would be able to do so by simply selecting a different ET source on their DSS front-end without any knowledge of how that source were connected to their DSS.

SWE deals only with data from sensors, however there are many non sensor-derived data sets that may be able to be used by a DSS. To connect to such datasets and to make sense of the data they present would require a data interchange format similar to those provided by SWE. A specification that was a superset of SWE's sensor specifications that included other non-sensor-based data sources or a specification that was an addition to SWE's specifications is needed. This specification, whatever form it took, would describe non-biophysical data sources (such as economic data sources), calculated data sets and also historical data sets. This would then provide DSS designers with a comprehensive specification of existing and potential data sources that are or could be used by DSS. If new data sources were designed to present data in a standards compliant way, a DSS design may be able to pre-emptively allow for their use as they would be reasonably similar to existing data sources to connect to.

SWE uses Web Services protocols for connecting to data sources and sets over the internet. If both sensor-based and non sensor-based data were presented in a standards compliant manner, Web Services resources may all be accessed in a one way as if through a portal. Such direct connectivity of sensors to the internet and therefore to internet-enabled tools, such as DSS, allow for information sharing which may be used for benchmarking, ground truthing remote equipment and the interpolation of results to areas were there are no local sensors.

The authors believe methods for addressing the second obstacle, mentioned above, require a new form of DSS reasoning engine and will not be considered here.

2.2. Architecture to enable Benchmarking

Something that has not been done by current agricultural DSS of any category in Australia is to provide instances of decision support advice or the recorded outcomes of that advice's use to multiple users for group learning. If a DSS were *inherently connected* to a network or the internet it could provide this functionality. By the phrase '*inherently connected'* the author means 'designed with connectivity in mind for all aspects of the DSS operation". This is in contrast to the design of all extant agricultural DSS in Australia which mostly act as standalone software systems that, if connected to remote resources at all, use connectivity to achieve small, singular, tasks such as the retrieval or delivery of data only. An *inherently connected* DSS could, for example, collect data from multiple sources (perhaps UDDS and Web Services resource via a portal) and store it remotely to the DSS user, perhaps on a server. It could then run models and compute derived personalised and non-personalised data, also on a server, and then store that data, along with the user's usage metrics to be used in decision support for other users.

Such a DSS would, in many senses consider system users as data sources rather than users and in doing so could be enabled to learn from such users by pattern recognition techniques as is currently done with standard data sources. In addition to benchmarking, such architecture would allow for:

1. New science, in both the agricultural sciences and information engineering fields, to be added to the DSS without a new product or version release (this would be achieved in the same way that Microsoft upgraded Windows XP with the addition of 'service packs' downloadable over the web, rather than requiring customers to purchase new full versions).
2. The possibility for accumulated usage data to be analysed by researchers that may then be fed back into the DSS to further enhance decision support or used elsewhere.
3. The possibility for irrigation decisions on different scales to benefit from a single information repository, for example irrigation companies producing better estimates of their growers' needs thought the enhanced ability to monitor irrigation demands.

3. NEW FRONT-END SYSTEMS

3.1. Modular Design

A concept of modular presentation, whereby base data, from sensors and non sensors, data from models and datasets derived from the base data was available for independent presentation, would allow a user to customise the decision support they received. Such modularisation could possibly be achieved by designing a display 'shell' that could then be altered for many sorts of data. Users could perhaps even create their own module, from such a shell which may provide the front-end to UDDS.

A website that has made some progress towards displaying any datasets that any user may upload is Swivel.com. This website, while not a scientific research tool, nonetheless leads the way in terms of heterogeneous data fusion. An agricultural DSS could emulate some aspects of Swivel.com in attempts to provide for the display of different types of data.

3.2. AJAX and cached presentation

AJAX, an acronym for Asynchronous JavaScript And XML it is an approach to client-server programming for internet resources that uses the power of modern internet browsers to access server data 'asynchronously', that is at times other than client request times (Garrett 2005). This is in contrast to the approach taken by most web page designers including DSS designers that use web-based presentation which typically sees a new page loaded when requested by a user through the internet browser on the client computer. This approach allows applications viewed thought internet browsers to act similarly to desktop applications by updating parts of the screen without reloading the entire screen. This approach to web design is used extensively by companies such as Microsoft Inc. and Google Inc.

The authors postulate that a DSS running calculations on a server remote to the DSS user could use AJAX methodology to present numerous data sets and modelled scenarios to a user, seemingly instantaneously. For example, a user might wish to firstly view modelled irrigation event timings that maximise water use efficiency and then secondarily view modelled irrigation event timings that favour a reduction in electrical power costs at the expense of water use efficiency. To seamlessly generate both sets of outcomes, at run-time, may not be possible for a desktop computer due to performance limitations but may be possible on a very fast server or server cluster. Displaying both those outcome sets seemingly instantaneously would not be possible using standard internet design techniques but would be so using the AJAX approach.

4. CONCLUSION

A new generation of decision support systems can be conceptualised for use in irrigation in Australia. Such a generation of DSS would hopefully be better placed to address some of the large problems facing DSS in agriculture today, particularly that of poor uptake by allowing for greater flexibility in the way data is presented and information that is displayed. By allowing for greater user control of the DSS interface and data sources used, the DSS should be able to address more of a particular users' wants and this may encourage uptake. Potentially such a customisable DSS could also be run by a by a consultant on behalf of a n irrigator and tailored to that irrigator's needs, rather than the consultant relying on interpreting results for that irrigator from a 'one size fits all' DSS.

Some properties of new generation DSS are based on technologies, such as Web Services, that have not yet seen use in the agricultural DSS arena and are thought to be readily achievable, while other potential properties need further informatics investigation and research to achieve realisation.

5. REFERENCES

Forrest, B. (2006). Controversy about our "Web 2.0" service mark. Retrieved July 2007 http://radar.oreilly.com/archives/2006/05/controversy_about_our_web_20_s.html

Gachet, A. and P. Haettenschwiler (2003). "A jini-based software framework for developing distributed cooperative decision support systems " Softw. Pract. Exper. **33**(3): 221.

Garrett, J (2005) Ajax: A New Approach to Web Applications http://adaptivepath.com/publications/essays/archives/000385.php. Adaptive Path online publications.

Hayman, P. (2004). Decision support systems in Australian dryland farming: A promising past, a disappointing present and uncertain future. 4th International Crop Science Congress. Brisbane, Australia, Crop Science Society of America.

Hornbuckle, J. W., E. W. Christen, et al. (2005). Use of SIRMOD as a Quasi Real Time Surface Irrigation Decision Support System. MODSIM 2005 International Congress on Modelling and Simulation. Melbourne, Australia, Modelling and Simulation Society of Australia and New Zealand.

Inman-Bamber, G. and S. Attard (2005). Inventory of Australian Software Tools for On Farm Water Management, CRC-IF.

Inman-Bamber, N. G., W. A. Webb, et al. (2006). "Participatory irrigation research and scheduling in the Ord: R and D." PROCEEDINGS- AUSTRALIAN SOCIETY OF SUGAR CANE TECHNOLOGISTS **Year 2006**: 155-163.

Lisson et al., 2003 S.N. Lisson, L.E. Brennan, K.L. Bristow, B.A. Keating and D.A. Hughes, DAMEA$Y–Software for assessing the costs and benefits of on-farm storage based production systems, *Agr. Syst.* **76** (2003), pp. 19–38.

Open Geospatial Consortium. (2007). "Sensor Web Enablement." Retrieved July 2007, http://www.opengeospatial.org/projects/groups/sensorweb.

Power, D. (2007). "Ask Dan! about DSS.", Retrieved July 2007 from http://dssresources.com/faq/index.php?sid=28928&lang=en&.

Sprague, R. H. (1980). "A framework for the development of decision support systems." MIS Quarterly **4**(4): 1–26.